Natural gas prices with Twitter sentiment scores

About Dataset Context: Natural gas is a fossil energy source that formed deep beneath the earth's surface. Natural gas contains many different compounds. The largest component of natural gas is methane, a compound with one carbon atom and four hydrogen atoms (CH4). Natural gas also contains smaller amounts of natural gas liquids (NGLs, which are also hydrocarbon gas liquids), and nonhydrocarbon gases, such as carbon dioxide and water vapor. We use natural gas as a fuel and to make materials and chemicals.

Henry Hub spot price: The Henry Hub pipeline is the pricing point for natural gas futures on the New York Mercantile Exchange. The NYMEX contract for deliveries at Henry Hub began trading in 1990 and is deliverable 18 months in the future. The settlement prices at Henry Hub are used as benchmarks for the entire North American natural gas market and parts of the global liquid natural gas (LNG) market.

Henry Hub is an important market clearing pricing concept because it is based on the actual supply and demand of natural gas as a stand-alone commodity. Other natural gas markets like Europe have fragmented hub pricing points. This means natural gas prices are often indexed to crude oil, which can have very different supply and demand factors affecting its price. Attempts are being made to develop European hub pricing points in the Netherlands and the UK, but this has proved difficult so far due to competition from national hubs. Asian natural gas markets are even more fragmented and have no defined hub pricing point, although Singapore would like to serve this regional role. Consequently, all Asian natural gas prices are either indexed to crude oil or linked to Henry Hub.

About the data: The data is a daily entry of the Henry hub spot price from 2015 to 2022 along with the sentiment scores of the day collected by analyzing relevant tweets on the topic.

Acknowledgement: This data is provided by scraping the website: https://eia.gov/ All credits and rights belong to them.

In [ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import calendar
```

In [13]:
```
import pandas as pd
data=pd.read_csv('https://raw.githubusercontent.com/gayathriravi2111997/project-try/main/data.csv')
data.head()
```

Out[13]:

	Unnamed: 0	Day	Price	Sentiment
0	0	20150105	3.22	-0.534077
1	1	20150106	2.98	-0.337025
2	2	20150107	3.08	-0.065357
3	3	20150108	2.92	-0.326590
4	4	20150109	2.96	-0.374404

In [14]:
```
data.tail()
```

Out[14]:

	Unnamed: 0	Day	Price	Sentiment
1869	1869	20220421	6.88	-0.583935
1870	1870	20220422	6.59	-0.402193
1871	1871	20220425	6.42	-0.403045
1872	1872	20220426	6.89	-0.401374
1873	1873	20220503	7.84	-0.073315

In [16]:
```
data
```

Out[16]:

	Unnamed: 0	Day	Price	Sentiment
0	0	20150105	3.22	-0.534077
1	1	20150106	2.98	-0.337025
2	2	20150107	3.08	-0.065357
3	3	20150108	2.92	-0.326590
4	4	20150109	2.96	-0.374404
...	...	...	...	...
1869	1869	20220421	6.88	-0.583935
1870	1870	20220422	6.59	-0.402193
1871	1871	20220425	6.42	-0.403045
1872	1872	20220426	6.89	-0.401374
1873	1873	20220503	7.84	-0.073315

1874 rows × 4 columns

In [11]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 0 entries
Columns: 1286 entries, {"metadata":{"kernelspec":{"language":"python" to metadata:{}}]}.24
dtypes: object(1286)
memory usage: 0.0+ bytes
```

In [17]:
```
data.describe()
```

Out[17]:

	Unnamed: 0	Day	Price	Sentiment
count	1874.000000	1.874000e+03	1874.000000	1874.000000
mean	936.500000	2.018235e+07	2.918623	-0.299520
std	541.121521	2.119460e+04	1.050903	0.192107
min	0.000000	2.015010e+07	0.000000	-0.819287
25%	468.250000	2.016103e+07	2.390000	-0.435230
50%	936.500000	2.018082e+07	2.790000	-0.313156
75%	1404.750000	2.020062e+07	3.070000	-0.158241
max	1873.000000	2.022050e+07	23.860000	0.693440

In [18]:
```
data.isnull().sum()
```

Out[18]:
```
Unnamed: 0    0
Day           0
Price         0
Sentiment     0
dtype: int64
```

In [19]:
```
data.shape
```

Out[19]:
```
(1874, 4)
```

In [25]:
```
data.nunique()
```

Out[25]:
```
Unnamed: 0    1874
Day           1874
Price          363
Sentiment      363
dtype: int64
```

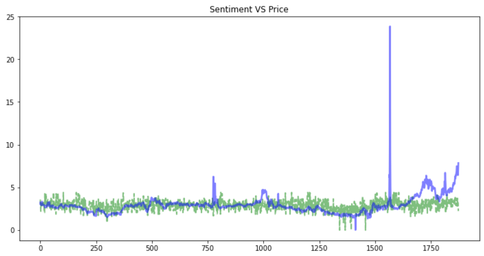
In [26]:
```
data.Price.unique()
```

Out[26]:
```
array([ 3.22,  2.98,  3.08,  2.92,  2.96,  2.9 ,  3.15,  3.32,  3.11,
        2.94,  2.95,  2.89,  2.88,  2.67,  2.73,  2.66,  2.62,  2.86,
        2.75,  3.02,  3.21,  2.79,  3.  ,  3.27,  2.76,  2.82,  2.72,
        2.68,  2.8 ,  2.77,  2.64,  2.65,  2.63,  2.71,  2.58,  2.61,
        2.57,  2.59,  2.5 ,  2.55,  2.56,  2.74,  2.78,  2.85,  2.87,
        3.01,  3.04,  2.6 ,  2.81,  2.93,  2.83,  2.7 ,  2.84,  2.91,
        2.69,  2.47,  2.37,  2.34,  2.35,  2.49,  2.48,  2.54,  2.42,
        2.43,  2.46,  2.28,  2.18,  2.14,  2.15,  1.98,  1.92,  2.04,
        2.19,  2.13,  2.05,  2.02,  2.1 ,  2.11,  2.06,  2.01,  2.  ,
        1.95,  1.79,  1.7 ,  1.66,  1.74,  1.76,  1.73,  1.63,  2.39,
        2.33,  2.4 ,  2.21,  2.23,  2.22,  2.16,  2.26,  2.07,  2.08,
        2.17,  1.93,  1.89,  1.83,  1.86,  1.85,  1.78,  1.62,  1.57,
        1.6 ,  1.59,  1.49,  1.56,  1.61,  1.72,  1.68,  1.81,  1.77,
        1.84,  1.87,  1.88,  1.96,  1.91,  1.99,  1.9 ,  1.97,  1.71,
        1.94,  2.03,  2.12,  2.25,  2.3 ,  2.31,  2.32,  2.36,  2.97,
        3.07,  3.12,  3.19,  3.18,  3.06,  3.09,  3.17,  3.16,  3.25,
        2.53,  2.41,  2.2 ,  3.44,  3.75,  3.8 ,  3.68,  3.6 ,  3.65,
        3.55,  3.56,  3.51,  3.39,  3.5 ,  3.7 ,  3.71,  3.41,  3.42,
        3.38,  3.14,  3.36,  3.37,  3.26,  3.23,  3.31,  3.13,  3.03,
        3.05,  2.52,  2.44,  2.99,  3.2 ,  3.1 ,  3.24,  3.69,  6.24,
        4.65,  4.06,  5.46,  3.92,  3.35,  3.54,  3.58,  3.34,  3.28,
        3.4 ,  3.45,  3.3 ,  3.53,  3.74,  3.96,  4.1 ,  4.69,  4.3 ,
        4.7 ,  4.28,  4.5 ,  4.61,  4.4 ,  4.44,  4.51,  4.54,  4.53,
        4.31,  3.99,  3.64,  3.43,  4.25,  2.45,  2.38,  2.27,  2.29,
        2.09,  2.24,  2.51,  1.75,  1.82,  1.8 ,  1.65,  1.69,  1.55,
        1.5 ,  1.64,  1.58,  1.67,  1.43,  1.51,  1.53,  1.42,  0.  ,
        1.33,  1.41,  3.49,  3.76,  6.5 ,  6.12, 11.32, 23.86,  8.56,
        4.96,  3.62,  3.79,  3.67,  3.66,  3.78,  3.82,  3.94,  4.02,
        4.11,  4.09,  4.15,  4.03,  4.2 ,  4.27,  4.21,  4.24,  4.12,
        4.07,  3.95,  3.93,  3.86,  3.83,  4.35,  4.33,  4.45,  4.77,
        4.71,  4.66,  4.97,  5.13,  5.21,  5.39,  5.66,  5.52,  5.32,
        5.25,  4.92,  4.94,  5.1 ,  5.53,  5.94,  5.73,  5.58,  5.61,
        5.8 ,  6.37,  6.  ,  5.71,  5.34,  5.56,  5.92,  5.44,  5.01,
        4.81,  4.87,  5.72,  5.59,  5.91,  5.68,  5.49,  5.22,  5.33,
        5.51,  5.08,  4.56,  5.11,  4.82,  4.95,  4.9 ,  4.83,  4.93,
        4.52,  4.08,  4.05,  3.91,  3.73,  4.16,  4.62,  4.78,  4.37,
        4.55,  4.89,  4.43,  5.69,  5.45,  6.7 ,  5.84,  4.13,  4.04,
        4.39,  4.57,  4.48,  4.59,  4.63,  4.46,  4.36,  4.74,  4.79,
        4.68,  4.8 ,  5.  ,  5.26,  5.19,  5.43,  5.95,  6.29,  6.05,
        6.38,  6.35,  6.59,  6.68,  6.94,  7.48,  7.44,  7.12,  6.88,
        6.42,  6.89,  7.84])
```

In [28]:
```
data.Price.value_counts()
```

Out[28]:
```
2.88    34
2.75    32
2.89    29
2.76    29
2.95    25
        ..
5.39     1
5.21     1
5.13     1
4.66     1
7.84     1
Name: Price, Length: 363, dtype: int64
```

In [31]:
```
import matplotlib.pyplot as plt
avg = sum(list(data.Price))/len(list(data.Price))
sen = list(data.Sentiment)
price = list(data.Price)
for i in range (len(sen)):
    sen[i] = -(sen[i]*avg)+2
plt.figure(figsize=(12, 6))
l1 = plt.plot(sen, lw = 2, color='green', alpha=0.5, linestyle='dashed')
l2 = plt.plot(price, lw = 2.5, color='blue', alpha=0.5)
plt.title('Sentiment VS Price')
plt.show()
```



In [ ]:
```
```